Analyzing Technology Interactions

Chirag B. Patel*

Prof. Dimitri N. Mavris†

Aerospace System Design Lab, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.

Technology selection is a crucial step in the design of new complex systems. When many technologies are to be selected from a large pool of available technologies, it is very important that the interactions among these selected technologies are accounted for while assessing their impact on the system. This paper will discuss some of the intricacies involved in technology interactions and current method of *Technology Constraint Matrix* used to account for them. The advantages and limitations of this method are discussed and a new approach to analyze technology interactions based on the principles of *Graph Theory* is introduced.

Nomenclature

$TIES$	Technology Identification Evaluation and Selection
GA	Genetic Algorithm
GT	Graph Theory
$V(G)$	Set of Vertices of Graph G
$E(G)$	Set of Edges of Graph G
$T(V, E)$	Technology graph
t	Number of Available Technologies
e	Number of Constraints among Technologies
i_G	Number of Independent Sets for Graph G
I_{G^p}	Average Number of Independent Sets for Random Graph G^p

I. Introduction

Technology interaction analysis is an important aspect of a technology selection process for large scale complex systems. Technologies can interact with each other in a variety of ways and are manifested in the form of their impact on the system. A modern commercial aircraft is one such system that has many technologies coming together. Selection of technologies for this system is a complex task. It is important to ensure that there are no conflicting or incompatible technologies present in the group of technologies selected, which are selected on the basis of their impact to performance and economic parameters in consideration. This is a combinatorial optimization problem where the problem size geometrically increases with an increase in the number of technology options available. Technology interactions act as a constraint in this combinatorial optimization and tend to reduce the total number of permissible combinations and make the entire search space more complex.

A. TIES with GA

When the number of technologies available for the system are more than about 15, it is not wise to search the entire combinatorial space for the best solutions. To handle the search in this combinatorial space, an

*Graduate Research Assistant, Student Member AIAA

†Director and Boeing Professor of Advanced Aerospace Systems Analysis, Associate Fellow AIAA

approach has been developed, which combines a GA in conjunction with the TIES process.[1] TIES is a comprehensive and structured method to allow for designs of complex systems which result in high quality and competitive cost to meet future, aggressive customer requirements. The basic theory behind the TIES method has been extensively explained by Mavris[2] and Kirby.[3] TIES provides a framework to evaluate impact of technologies on the system level metrics. A GA is wrapped around this framework to evaluate a large number of technology combinations in short time and select the best set of combinations. GA works by creating a pool of technology combinations and evaluating them in the technology impact model. This yields estimates of how each combination impacts the entire system performance.

To get an accurate estimate of the impact of technologies on system level metrics, it is necessary to consider the interactions among these technologies. It is relatively easy to account for basic compatibility relations but when the interactions are *non-simple* the problem becomes more complex.

II. Technology Interactions

Various types of interactions or relations exists among technologies. An initial attempt to model technology interactions in the context of technology selection for preliminary aircraft design is described by Kirby.[4] In this treatment of interactions, physical compatibility/incompatibility rules between technologies are formalized in the form of a *Technology Compatibility Matrix*. Roth and Patel[5] categorize various types of interactions that exist among technologies into two main groups: *Simple Interactions* and *Non-Simple Interactions*.

A. Simple Interactions

Simple technology interactions are boolean relationships among technologies. The basic types of boolean technology interactions are shown in Figure 1. The most likely relationship that exists among technologies is of independence. That is a technology is completely independent of the rest and can be used with any other technology. In other words, it is compatible with all the technologies and does not interact with any other. The next is incompatibility where a technology is not compatible with another and the two cannot be used together. Hence, either technology a OR b has to be used. Incompatibilities arise when two technologies are competing for the same function or when one technology severely degrades the functionality of another. For example, there can be two structural technologies such as composites and integrally stiffened aluminium for construction of wings and only one can be used. As this relationship is symmetric it can be accounted by using only the super diagonal elements of a $n \times n$ (square) matrix as shown in equation 1. Here, for any i, j such that $1 \leq i \leq j \leq t$, if technology i and j are incompatible, then $c_{i,j} = 1$, otherwise 0.

$$C = \begin{pmatrix} 0 & c_{1,2} & \dots & c_{1,t} \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & c_{t-1,t} \\ 0 & 0 & \dots & 0 \end{pmatrix} \tag{1}$$

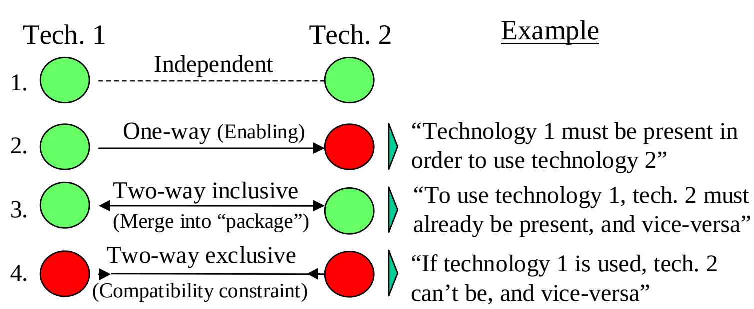


Figure 1. Simple Technology Interactions (Adapted from [1])

Another form of boolean interaction that can be present among technologies is an *Enabling* relationship. Here, the presence of one technology is necessary for proper functioning of the other, therefore, technology a

AND b have to be used together. Enabling relationship is not symmetric and can act in two directions. Either a can be an enabling technology for b i.e. a can work independently while b cannot work without a, or vice versa. There can also be a much stronger relationship where neither a nor b can work independently. In this case these two technologies can be merged into a "package". For enabling interactions, as the relationship is not symmetric, both the sub and super diagonal elements of a $t \times t$ matrix are required to define the interactions as shown in equation 2. In this formulation, for any i, j such that $1 \leq i, j \leq t$, if i is an enabling technology for j and i is independent of j then $e_{i,j} = 0$ and $e_{j,i} = 1$. [a] If both i and j are enabled by each other then $e_{i,j} = 1$ and $e_{j,i} = 1$.

$$E = \begin{pmatrix} 0 & e_{1,2} & \cdots & e_{1,t} \\ e_{2,1} & 0 & \ddots & \vdots \\ \vdots & \ddots & 0 & e_{t-1,t} \\ e_{t,1} & \cdots & e_{t,t-1} & 0 \end{pmatrix} \tag{2}$$

1. Technology Constraint Matrix

While implementing simple technology interactions in the TIES methodology, the compatibility relationship in form of equation 1 and enabling relations in form of equation 2 are combined in a *Technology Constraint Matrix*. It is possible to combine the two equations into one because two relationships are mutually exclusive. That is to say that when two technologies are incompatible, they cannot be enabling each other at the same time and vice versa. Here, the enabling technology relationship is denoted by -1 instead of 1 as it conflicts with the notation of incompatibility relationship. A notional technology constraint matrix as implemented while selecting technologies using GAs is listed in Table 1.

Table 1. Technology Constraint Matrix

	T1	T2	T3	T4	…
T1	0	1	0	-1	
T2	0	0	-1	0	
T3	0	0	0	0	
T4	-1	0	0	0	
⋮					⋱

B. Non-Simple Interactions

Simple technology interactions as described before are primarily boolean relationships. Here, the impact of technology interactions on system level metrics is additive. That is when two technologies enable each other and are considered together for a technology combination, their combined impact on a system level metric is the sum of each technology considered individually. When the technologies are incompatible, only one technology can be considered at a time. This assumption is a vast simplification and generally not valid when real cases are considered. It considerably limits the technology combinatorial space. There can be various levels of interactions between two technologies rather than just -1, 0 and 1 as denoted in TCM. These type of interactions are called non-boolean interactions.[5] Significant effort is required for modeling this type of interactions.

Various types of more complex boolean interactions also arise among technologies. A simple example is a three way interaction arising among three technologies. If the technologies are independent, there are 8 permissible combinations. However, if all are incompatible with each other, three technologies can only be used independently or none is used, i.e. 4 permissible combinations. In general, it not easy to count the exact amount of permissible technology combinations. Principles of *Graph Theory* can help us enumerate permissible combinations and better understand the technology combinatorial space. Graph Theory is an area of discrete mathematics and the relation of technology interactions with this field is explored in following section.

[a] $e_{i,j}$ is read as i is enabled by j.

III. Graph Theory Connection

A *graph* is a triple consisting of a vertex set, $V(G)$, an edge set, $E(G)$ and a relation that associates with each edge two vertices (not necessarily distinct) called its endpoints.[6] A graph can be used to represent a technology space, vertices represent the technologies and edges represent the interaction between two *distinct* technologies. For now, we denote non-directional edges between technology vertices and these edges represents incompatibility relations. A notional technology interaction space is shown in the form of a graph in figure 2. Here, T3 and T8 do not have any edges incident to them, hence they are totally independent technologies. However, for example, T1 has two incident edges and therefore it is incompatible with two technologies namely T7 and T4.

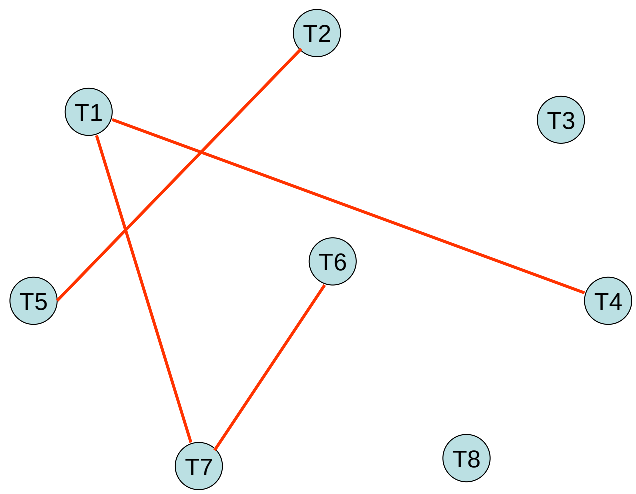


Figure 2. Technology Graph T

A. Counting the Permissible Technology Combinations

Permissible technology combinations are sets of technologies that do not violate any incompatibility or enabling constraints. These sets do not have any incompatible technologies within them and they also respect the enabling constraints. Even considering only the incompatibility constraints, it is difficult to quantify or enumerate the number of permissible technology combinations. Graph theory can help us to tackle this problem. As mentioned before, the technology space is seen as a graph with technologies as vertices and non-directional edges as incompatibility constraints. The maximum number of edges a graph can have is given by:

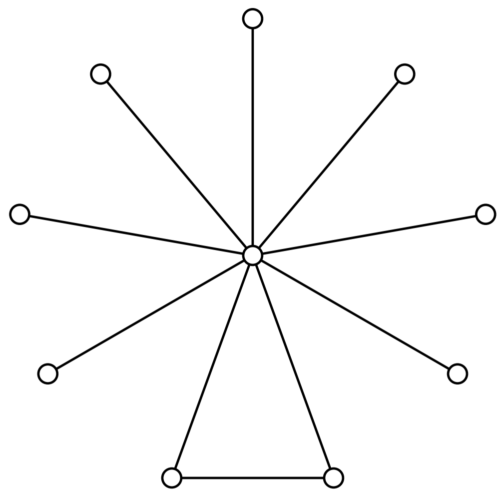
$$\binom{t}{2} = \frac{t}{2}(t-1)$$

This is equivalent to the maximum number of incompatibilities a group of technologies can have among themselves, and in such a situation, each technology can be used individually or none at all. Therefore, the maximum number of permissible combinations here will be $t+1$. When all the technologies are independent and there are no edges between them, the maximum number of permissible combinations is 2^t.

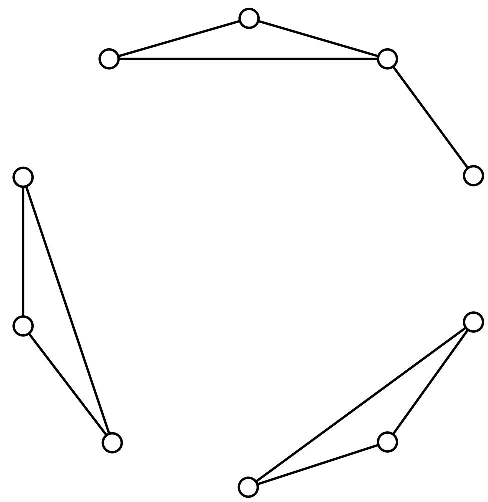
The number of permissible combinations to be counted in the above mentioned extreme cases is trivial, but it is a difficult problem when the number of incompatibilities is between 0 and $\binom{t}{2}$. In graph theoretic parallels, the problem is to find total number of independent sets. A subset S of $V(G)$ is called an *independent set* of G if no two vertices of S are adjacent[b] in G.[7] In the literature, independent sets are also know as stable sets or cocliques. The number of independent sets in T not only depends on the number of vertices and edges but also on the arrangement of edges between the vertices. For example, different arrangements of 10 incompatibilities among 10 technologies that give maximum and minimum number of independent sets possible is shown in figure 3 on the following page. The maximum number of independent sets are obtained

[b]Two vertices are *adjacent* if there is an edge between them.

when one technology is incompatible with all other technologies and the remaining are as independent among themselves as possible. In other words, one vertex has maximum degree[c], $t-1$ in this case, and the remaining vertices have minimum possible degrees. This arrangement is demonstrated in figure 3(a) and the vertex degrees are $[9, 2, 2, 1, 1, 1, 1, 1, 1, 1]$. On the other hand, minimum number of independent sets are obtained when the technologies form groups or components that are complete graphs in themselves, i.e. all the technologies within a component are incompatible with each other. This arrangement is represented in figure 3(b) with 3 triangles and the remaining vertex attached to one of the triangles. The above observations are made using an integrated environment for graph theory called *newGRAPH*.[8]



(a) Maximum 384



(b) Minimum 111

Figure 3. Permissible Combinations with $t = 10$ and $e = 10$

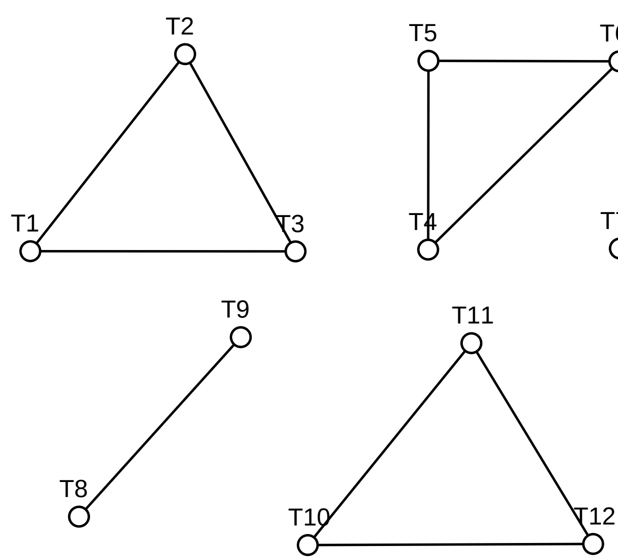


Figure 4. 12 Interacting Technologies From Total of 29

While analyzing real technologies, one finds majority of them are independent and the remaining are not completely interconnected but form small components of mutually interacting technologies. This fact

[c]The *degree* of vertex is the number of incident edges

proves to be helpful while calculating the total number of permissible combinations as the problem of enumerating the independent sets of a large connected graph is difficult and computationally intensive. Let us consider a real example with 29 technologies, out of which, 17 are totally independent and 12 technologies have 11 incompatibility constraints among them as depicted in figure 4. This graph has four disconnected components. Here each component has a maximum 4 vertices and it is easy to manually count the number of independent sets for each component. Now, let a and b denote two components with i_a and i_b number of independent sets (not counting the null set) respectively. With basic combinatorics, when these two components are included in a single graph, i.e. union of two components, the total number of independent sets of $a+b$ is given by the following relation:

$$i_{a+b} = i_a \times i_b + i_a + i_b$$

In general, for a graph G with w components, the number of independent sets is given by equation 3. In many examples, these components are complete graphs or cliques, i.e., each technology is incompatible with every other technology in the component. In such cases, the number of independent sets for the components is same as their cardinality and equation 3 becomes similar to the one described by Utturwar et. al.[9]

$$i_G = \prod_{j=1}^{w} (i_j + 1) - 1 \tag{3}$$

Now, the number of independent sets along with the null set for a union T of graph G having i_G independent sets and k independent vertices is given by equation 4.

$$i_T + 1 = 2^k \times (i_G + 1) \tag{4}$$

Applying equation 3 for the four components of figure 4, the number of independent sets we get is 335. Considering the remaining 17 independent technologies and applying equation 4, the total number of permissible technology combinations are $44,040,192$ (including the null set). This is out of $2^{29} = 536,870,912$ possible combinations. Thus, over 90% of the total technology combinations become impermissible by only about 2.7% of the total possible edges.

B. Average Number of Independent Sets

Before investing the time and resources to precisely enumerate permissible combinations, it is useful to know the average number of independent sets a technology graph can have. Random graphs and associated probabilistic techniques are useful for this type of analysis as illustrated by Wilf.[10] Let us consider a random graph $G^p(n,p)$ with n vertices and p is the probability with which each of the $\binom{n}{2}$ edges occur independently. If $S \subseteq V(G^p)$, then the average number of independent sets is the sum of the probability that every S is independent, over all the vertex subsets S. If S has m vertices, then the probability that S is independent is same as the probability that there are no edges among m vertices of S. With $(1-p)$ probability of absence of edge between two vertices and $m(m-1)/2$ edges possible in S, the expression for average number of independent sets is given by equation 5.

$$I_{G^p} = \sum_{m=0}^{n} \binom{n}{m} (1-p)^{m(m-1)/2} \tag{5}$$

For the notional example with 10 technologies and 10 incompatibilities or edges of figure 3 on the previous page, the fraction of edges present out of total possible 45 is $10/45 = 0.2222$. Applying equation 5 with $n = 10$ and $p = 0.2222$ we get $I_{G^p} = 174.88$. This number is closer to the lowest possible value of 111 than the maximum number of 384 because there are more arrangements of edges on a random graph that result in the values closer to the minimum than the ones that result in the values closer to the maximum. For the example with 29 technologies and 11 edges, $I_{T^p} = 5.2 \times 10^7$ and the actual number of combinations as counted in the previous section is about 4.4×10^7. Thus, whenever the technologies interact within small groups and these groups are almost complete graphs, the number of permissible combinations can be significantly lower than the average number of independent sets of corresponding random graph.

C. Enumeration with Backtracking

Previous results show that the average number of independent sets can be considerably smaller than 2^t for certain types of technology graphs. This average number give an upper bound for the number of permissible technology combinations. Hence, if I_{T^p} is within the limits of available computation resources, it may be feasible to perform a full factorial analysis instead of going for a stochastic approach, as mentioned earlier, using GAs. Now, to perform a full factorial analysis, it is necessary to enumerate all the permissible combinations of technologies. A prevalent search technique called *backtracking* is described that can be used to enumerate the permissible combinations. This technique is generally used to solve graph theoretic problems such as finding maximum independent set or clique,[11] graph coloring, etc. Backtracking essentially performs a depth first search on the technology graph.

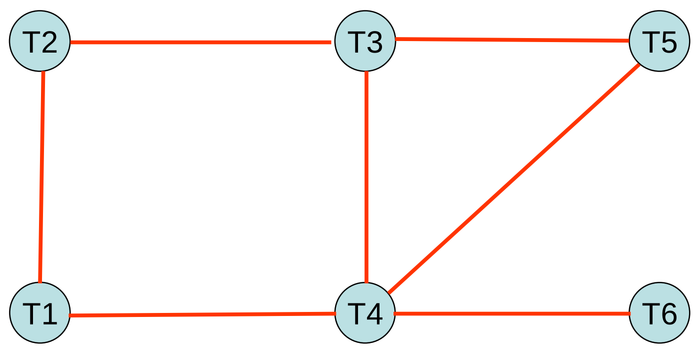


Figure 5. Graph G for backtracking

Consider a graph G with 6 vertices and 7 edges as shown in figure 5. Starting with the first vertex, the independent set is $S := \{T1\}$. Now, we attempt to enlarge S and the next vertex we can add is T3 as T2 is connected to T1. The S now has $\{T1, T3\}$. After T3 we can only add T6 and cannot go any further, S is $\{T1, T3, T6\}$. Therefore, we backtrack one step at a time till we can find more options. In this example, we have to go back to T1 (delete T3 and T6 from S) and search for the next vertex that can be added, here it is T5. When all options are exhausted with T1, we start the process again with the next vertex and $S := \{T2\}$. A list of independent sets for the example as obtained by backtracking method is enumerated below.

$\{T1\}, \{T1, T3\}, \{T1, T3, T6\}, \{T1, T5\}, \{T1, T5, T6\}, \{T1, T6\}$

$\{T2\}, \{T2, T4\}, \{T2, T5\}, \{T2, T5, T6\}, \{T2, T6\}$

$\{T3\}, \{T3, T6\}$

$\{T4\}$

$\{T5\}, \{T5, T6\}$

$\{T6\}$

As observed before, the technology space for real problems is composed of small disjoint components and other independent technologies. Independent sets in each of these components can be enumerated using backtracking technique. In the technology evaluation environment of TIES with GA, a technology combination is represented by a row vector of zeros and ones; for e.g., a combination of T1, T3 and T6 in a graph with 6 technologies is represented as $[1, 0, 1, 0, 0, 1]$. A set of all permissible combinations in a component with n technologies is in the form a $i \times n$ matrix, with each row representing an independent set. The matrix of permissible combinations for n independent technologies is basically a binary conversion of a row of numbers from 0 to $2^n - 1$, with 2^n rows and n columns. Now, with matrices of permissible sets for all the components and independent technologies in place, the independent sets of the entire technology graph are enumerated using the logic behind equation 3 and 4. Consider two components a and b with independent set matrices of size $i_a \times j$ and $i_b \times k$ respectively. The independent sets for the union of a and b are obtained by concatenating each row of the first matrix with each of the other. This will result in a matrix of size $(i_a \times i_b) \times (j+k)$. This process is repeated till all the components and independent technologies are included.

D. Enabling Technologies

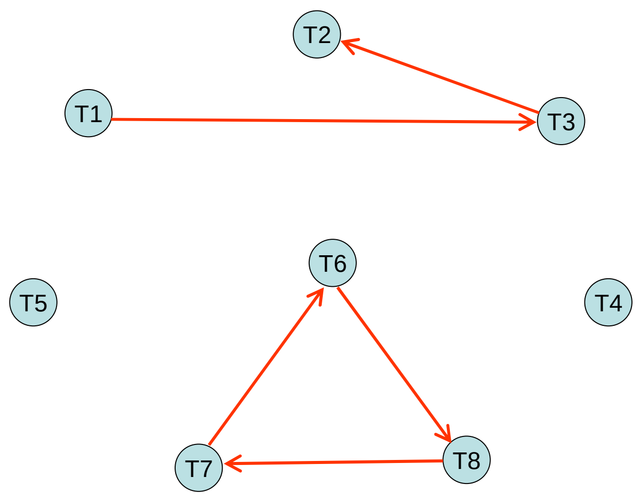


Figure 6. Digraph for Enabling Technologies

Observations made in previous sections consider only the incompatibilities in the technology space. There may be some technologies in the space that enable others and these can be visualized using graphs with directed edges known as digraphs as shown in figure 6. Here, the edges point towards the enabling technology, e.g., in figure 6, T1 is enabled by T3 and T3 in is enabled by T2. Hence, while T2 can function independently, T1 needs T3 and T3 needs T2 to function. Depending on the relationships, the complexity of this digraph may be reduced by merging some of the technologies. In figure 6, T6, T8 and T7 form a unidirectional cycle where one technology is enabled by the next. These can be merged into a single technology as none can function in absence of any other member of the cycle. This reduction can be adopted for any number of technologies as long as they form a unidirectional cycle and also for two mutually enabling technologies. Once the technology graph is reduced, backtracking technique can be applied with appropriate modifications to account for enabling relationships to enumerate the permissible combinations.

E. Example Problem

As mentioned before, TIES is the generic method used to create an environment to evaluate technology combinations. Once this environment is in place for a particular system, any number of technologies can be evaluated for that system, given the technology graph and *Technology Impact Matrix* (TIM) for those technologies. TIM represents the impact of each technology on certain key parameters known as technology metrics or k-factors.[12]

As an example, consider a fictitious problem with 17 aircraft technologies. These were created with 5 independent technologies combined with the 12 technology graph as shown in figure 4 on page 5 and a TIM was randomly generated with 10% nonzero values. The system under consideration is a commercial passenger aircraft whose 15 responses are tracked. The aim is to find a Pareto optimal technology combinations. The concept of Pareto optimality for technology selection is explained in detail by Patel et. al.[13]

The problem has 17 technologies and hence $2^{17} = 131072$ possible combinations. This is quite a high number to be evaluated on a PC. There are 11 edges out of 136 possible edges for 17 technologies. Now, the average number of independent sets or permissible technology combinations as given by equation 5 on page 6 is 15613. This number of combinations can be comfortably evaluated on a PC and for this the exact combinations have to be enumerated. Enumeration is accomplished using the backtracking and matrix concatenating technique described in the previous sections. The exact number of permissible combinations to be evaluated is 10752. It takes about 16 minutes for a 2 Ghz PC to evaluate these combinations. An efficient sorting algorithm is used to extract the Pareto optimal technology combinations out of 10752 points. Pareto front for response 1 and 2 is shown in figure 7. There are 195 technology combinations on the Pareto frontier of response 1 and 2. Pareto front solutions for more than two responses can also be searched with this sorting algorithm. In this example, the responses are standardized for Pareto sorting so that minimum values are better. For response 2, higher actual values are preferred and hence the negative sign. Each point on the plot corresponds to a particular response vector and technology combination.

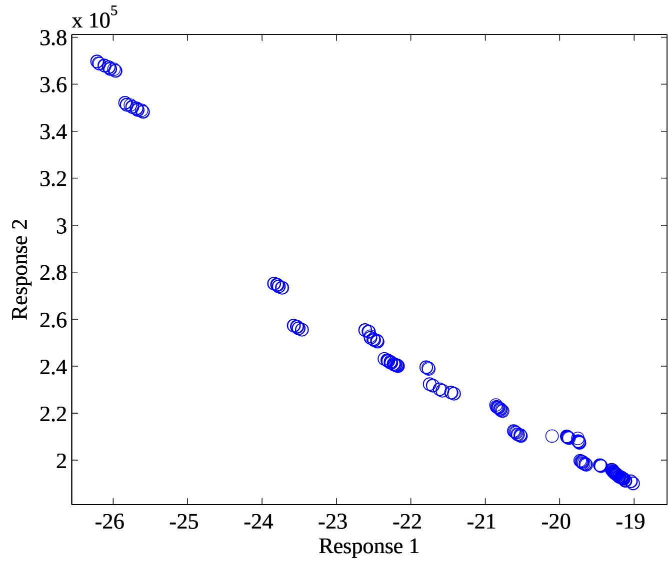


Figure 7. Pareto Optimal Technology Combinations

The decision makers can make tradeoffs on the Pareto frontier and select the most preferable technology combination. This exhaustive method helps decision makers take decisions based on all the information available. They can be confident of their decisions knowing that there can be no other points on the Pareto frontier apart from the ones displayed; as opposed to a stochastic method where only a subset of Pareto optimal solutions are available to the decision makers.[13]

IV. Technology Interactions with GAs

When the number of permissible technologies is too large for a full factorial analysis, a GA based approach is recommended. A few techniques have been developed in last five years that can be used to account for interactions while using GAs or other evolutionary algorithms in conjunction with TIES methods. These are part of two basic approaches.

A. Soft Constraints

This approach is a type of penalty method where the technology incompatibilities are treated as an objective function whose values are to be reduced through the generations of GA. Here, the incompatible technology sets may also be evaluated. This technique is employed by Roth and Patel[5] where incompatibility free final solution set were obtained with high enough weighting on the incompatibility constraints. The only information needed for this technique is the number of incompatibilities and enabling constraints present in certain set of technologies and there is no need to name the edges that cause those constraints. This number can be easily evaluated using adjacency matrix of the technology graph. For this two different matrices are created, one for incompatibilities and one for enabling. The adjacency matrix for technology graph with non-directional edges representing incompatibilities is a symmetric matrix where the $(j,k)^{th}$ entry represent the presence or absence of edge between vertices j and k. Matrix C of equation 1 is the upper triangular portion of the adjacency matrix. When the technology combination set is in the form of a $(1 \times t)$ vector S as shown before, it can be easily proved by basic algebra that the quantity $S \times C \times S^T$ gives the number of edges present in the technology set S.[d] For evaluating the number of enabling violations in S, adjacency matrix for the digraph is considered and this is same as matrix E of equation 2. In this case we are interested in the absence of directed edges in S and the number of enabling violations is given by the expression $S \times C \times \bar{S}^T$; here $\bar{S}$ denotes the vector S with all the ones changed to zeros and zeros to ones. If S is a $(n \times t)$ matrix for n technology combinations, above expressions can be used and the result of the product is a $n \times n$ matrix. The number of constraint violations for n combinations are found in n diagonal elements of the resultant matrix.

This technique for accounting interactions is very simple to implement with GAs. Its main drawback is that there is some probability that the final solution set has some incompatible combinations and to ensure that it is free of any constraint violations, the incompatibility and enabling functions evaluations have to be heavily weighted. Moreover, GA has to keep track of extra responses which has some degrading effect on its performance.

B. Hard Constraints

In this approach, the technology combinations that violate the incompatibility and enabling constraints are never included in the population pool of the GA. Raczynski et. al.[14] proposed a gene correction technique that allows only the compatible technology combinations to be evaluated by the optimizer. This algorithm detects incompatibilities in a technology set and removes certain technologies randomly from the set so that the resulting combination has all compatible technologies. This algorithm can be extended to search and repair for enabling technology combinations. It is included in the GA loop just before the reproduction operator so that no incompatible combination is evaluated. This technique has been shown to result in early convergence of function values as compared to the penalty method. It is flexible enough to be implemented for any type of technology graph.

The next technique that implements the hard constraint approach is the reduced bit system employed by Raczynski et.al.[15] When certain group of n technologies form a clique or complete graph among themselves, instead of 2^n combinations only $n+1$ can be used. Thus rather than using n columns or bits to represent n technologies, only $\lceil (ln(n+1)/ln(2)) \rceil$ bits may be used. Thus each combination of reduced bit system corresponds only to a compatible technology combination. When implemented for all the components of technology graph, reduced bit system eliminates the risk of creation of invalid combinations by the mutation and crossover operators of GA.

V. Conclusion

It can be observed from this study that technology interactions have significant effect on the technology combinatorial space. The principles of graph theory are shown to be very useful for analyzing the interactions and resultant technology combinatorial space. Technologies and interactions among them are analogous to the vertices and edges of graphs which are good visualizing tool for technology space. Random graphs provide important result that give an upper bound on the number of permissible technology combinations present in the technology space. Based on this number, decision can be made regarding a full factorial analysis or GA based approach for the problem. If full factorial analysis it to be performed, backtracking algorithm, which has its roots in graph theory, can be used to enumerate all the permissible combinations. If a GA based approach is preferred, technology interactions can be accounted by implementing some of the techniques described in the previous section.

The interactions considered for this paper are primarily boolean relationships. Here, technologies are either compatible or not, the impact of technologies on the system level metrics is additive. There are other non simple interactions possible that warrant further investigation. Digraphs with weighted edges may be used to model such interactions. These intermediate interactions are generally estimated by technology experts and hence their sensitivity on the results would have to be investigated. Graph theory and its concepts provide a good starting point for technology interaction analysis.

[d] S^T denotes transpose of S

References

[1]Roth, B., German, B. J., Mavris, D. N., and Macsotai, N., “Adaptive Selection of Engine Technology Solution Sets From a Large Combinatorial Space,” AIAA Paper 2001–3208, 2001.

[2]Mavris, D. N. and Kirby, M. R., “Technology Identification, Evaluation and Selection for Commercial Transport Aircraft,” SAWE Paper 2456, Category no. 11, 1999.

[3]Kirby, M. R. and Mavris, D. N., “A Technique for Selecting Emerging Technologies for a Fleet of Commercial Aircraft to Maximize R&D Investment,” SAE Paper 2001–01–3018, 2001.

[4]Kirby, M. R., *A Method for Technology Identification, Evaluation and Selection in Conceptual and Preliminary Aircraft Design*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA 30332, 2001.

[5]Roth, B. and Patel, C., “Application of Genetic Algorithms in the Engine Technology Selection Process,” *Journal of Engineering for Gas Turbines and Power*, Vol. 126, No. 4, 2004, pp. 693–700, also ASME Paper 2003–GT–38482, June 2003.

[6]West, D. B., *Introduction to Graph Theory*, Perintice-Hall Inc., 2nd ed., 2005.

[7]Bondy, J. A. and Murty, U. S. R., *Graph Theory with Applications*, American Elsevier Publishing Co., 1976.

[8]Stevanović, D., Brankov, V., Cvetković, D., and Simić, S., “newGRAPH: Integrated Enviornment for Graph Theory,” On-line resource: http://www.mi.sanu.ac.yu/newgraph/index.html.

[9]Utturwar, A., Rallabhandi, S., DeLaurentis, D., and Mavris, D. N., “A Bi-Level Optimization Approach for Technology Selection,” 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization AIAA-2002-5426, AIAA, Atlanta, GA, September 2002.

[10]Wilf, H. S., *Algorithms and Complexity*, Internet Edition, 1994, http://www/cis.upenn.edu/∼wilf.

[11]Bron, C. and Kerbosch, J., “Algorithm 457: Finding all Cliques of an Undirected Graph,” *Communications of the ACM*, Vol. 16, No. 9, 1973, pp. 575–577.

[12]Kirby, M. R. and Mavris, D. N., “An Approach for the Intelligent Assessment of Future Technology Portfolios,” Paper 2002-0515, AIAA, 2002.

[13]Patel, C. B., Kirby, M. R., and Mavris, D. N., “Niched-Pareto Genetic Algorithm for Aircraft Technology Selection Process,” 11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization AIAA-2006-6956, AIAA, Portsmouth, VA, September 2006.

[14]Raczynski, C. M., Lewe, J.-H., Kirby, M. R., and Mavris, D. N., “Technology Portfolio Assessments Using a Gene-Correction Genetic Algorithm,” AIAA Paper 2003–6731, 2003.

[15]Raczynski, C. M., Kirby, M. R., and Mavris, D. N., “Technology Portfolio Assessments Using a Multi-Objective Genetic Algorithm,” Tech. Rep. 2004-01-3134, SAE, Reno, NV, November 2004.